**WHAT IS CLAIMED IS:**

1. A graphical user interface (GUI) for configuring pipelines, the GUI displayable on a user computer monitor and comprising:

at least one pipe input set window configured to permit a user to define a type of pipe input set data;

at least one GUI page based on the type, the GUI page being generated by translating the type using a configuration file to a class and using Java reflection to generate an instance of the class, the instance producing the GUI page.

2. The GUI of Claim 1, wherein at least the pipe input set window and GUI page require no programming apart from an initial core code.

3. The GUI of Claim 1, wherein the GUI is an incremental GUI wherein GUI pages for new pipe components can be added incrementally without changing existing code.

4. The GUI of Claim 3, wherein at least one new pipe module is based on a pre-existing module type.

5. The GUI of Claim 3, wherein at least one new pipe module is based on a new user-defined component type.

6. The GUI of Claim 1, wherein the GUI defines a set of interfaces, each interface including plural functions, the GUI including a GUI representation part and a storage part, the GUI representation part defining how something is displayed and the storage part defining how GUI parameters are stored in an external storage.

7. The GUI of Claim 1, further comprising:

at least one *Pipe Output Set* tab for defining *PipeOutputSet* representative of a type of output data from the pipeline.

8. The GUI of Claim 1, further comprising:

at least one *Storage For TupleSets* tab for defining an arbitrary number of elements contained in a *StorageForTupleSets* component of the pipeline, individual input and output sets being definable for each element in the component.

9. The GUI of Claim 1, further comprising:

at least one *Pipe Modules* tab for defining an arbitrary number of *PipeModules* of the pipeline, a type being selected for each *PipeModule* using the tab, the type defining at least in part the GUI.

10. A graphical user interface (GUI) for a Pipeline architecture, comprising:

means for generating and modifying *Pipeline*s without writing any JAVA code apart from an initial core code.

11. The GUI of Claim 10, wherein the means includes:

at least one pipe input set window configured to permit a user to define a type of pipe input set data; and

at least one GUI page based on the type, the GUI page being generated by translating the type using a configuration file to a class and using Java reflection to generate an instance of the class, the instance producing the GUI page.

12. The GUI of Claim 11, wherein at least the pipe input set window and GUI page require no programming apart from an initial core code.

13. The GUI of Claim 11, wherein the GUI is an incremental GUI wherein GUI pages for new pipe modules can be added incrementally without changing existing code.

14. The GUI of Claim 11, wherein the GUI defines a set of interfaces, each interface including plural functions, the GUI including a GUI representation part and a storage part, the GUI representation part defining how something is displayed and the storage part defining how GUI parameters are stored in an external storage.

15. The GUI of Claim 11, further comprising:

at least one *Pipe Output Set* tab for defining *PipeOutputSet* representative of a type of output data from the pipeline.

16. The GUI of Claim 11, further comprising:

at least one *Storage For TupleSets* tab for defining an arbitrary number of elements contained in a *StorageForTupleSets* component of the pipeline, individual input and output sets being definable for each element in the component.

17. The GUI of Claim 11, further comprising:

at least one *Pipe Modules* tab for defining an arbitrary number of *PipeModules* of the pipeline, a type being selected for each *PipeModule* using the tab, the type defining at least in part the GUI.

18. The GUI of Claim 10, further comprising:

means for making available new pipeline module types without writing any JAVA code apart from an initial core code; and

means for adding a new type to a configuration file such that the new type is executable without recompiling the core code.

19. A method for generating a pipeline for processing data from at least one data store, comprising:

presenting a main GUI window;

using the main GUI window to access an initial core code;

using the main GUI window to access at least one subsequent GUI window; and

using the at least one subsequent GUI window to configure the pipeline at least in part.

20. The method of Claim 19, wherein the main GUI window is at least one pipe input set window configured to permit a user to define a type of pipe input set data, at least one GUI page based on the type being configurable.

21. The method of Claim 20, comprising:

generating the GUI page by translating the type using a configuration file to a class; and

using Java reflection to generate an instance of the class, the instance producing the GUI page.

22. The method of Claim 20, wherein the GUI defines a set of interfaces, each interface including plural functions, the GUI including a GUI representation part and a storage part, the GUI representation part defining how something is displayed and the storage part defining how GUI parameters are stored in an external storage.

23. The method of Claim 20, further comprising:

defining a representative of a type of output data from the pipeline.

24. The method of Claim 20, further comprising:

defining an arbitrary number of elements contained in a component of the pipeline, individual input and output sets being definable for each element in the component.

25. The method of Claim 20, further comprising:

defining an arbitrary number of *PipeModules* of the pipeline, a type being selected for each *PipeModule* using a tab, the type defining at least in part the GUI.